Semantics Based Intelligent Search in Large Digital Repositories using Hadoop MapReduce

Muhammad Idris[1], Shujaat Hussain[1], Taqdir Ali[1], Byeong Ho Kang[2] and Sungyoung Lee[1]

[1]Department of Computer Engineering, Kyung Hee University, Korea,
{idris, shujaat.hussain,taqdir.ali, sylee} @oslab.khu.ac.kr
[2]Dept. of Science, Engineering and Technology UoT, byeong.kang@utas.edu.au

Abstract. Information contained in large digital repositories consisting of billions of documents represented in various formats make it difficult to retrieve the desired information. It is necessary to develop techniques that are accurate and fast enough to retrieve the desired information from hay stack of online digital repositories. On one hand, Keyword based systems and techniques have high recall and performance, however, they have low precision. On the other hand, semantics based systems have high precision and good recall, however, their performance decreases with data growth. Therefore, to improve precision and performance, we propose semantics based searching framework using Hadoop MapReduce to process the data at large scale. We apply semantic techniques to extract required information from digital documents and MapReduce programming model to apply these techniques. Application of semantic techniques using MapReduce distributed model will result in high precision and good performance of user query result.

1 Introduction

Online digital repositories are increasing in size and varieties amounting to terabytes and petabytes of data[1]. The data growth in digital documents is even more than approximation of Moore's law. Performing search on large scale digital documents using traditional search engines is tiresome, slow and monotonous. The existing and established searching techniques use word-counting[2], and document indexing[3] to accomplish their job. Existing semantics systems performance decreases with the increase of information in large repositories. Numerous digital repositories such as IEEE[4], ACM[5] and MEDLINE[6] processed through distributed solutions available for traditional keywords and indexing techniques in[2][3], however, there is lack of a distributed system that process large information with its semantics and tagging efficiently. Conventional systems and techniques such as in [7][8][9][10] focus on processing data semantically to increase the accuracy of intended user query, however, its performance decreases because of its sequential and procedural nature. Searching data at large scale (big data) has been discussed by many researchers, many systems and techniques have been proposed using traditional sequential and distributed frameworks. MapReduce(MR) has also been adopted by many large organizations for

analytics such as in Google[11], Facebook[12] and XML processing[13], e-mails processing respectively. All these systems use Hadoop-MR, while no such system exists that adopts and considers semantics based searching of digital data using MR. In this paper, we propose Hadoop-MR[14] based parallel and distributed search technique in digital repositories with semantics as discussed in[7] to gain the accuracy and performance. Accuracy is achieved through adaption of semantic searching and performance through implementing in separate *map* and *reduce* phases as shown in Figure 1. Due to its distributed and independent nature of processing, the proposed framework outperforms existing semantics based processing systems.

2 Proposed Framework

In this section, we describe architecture of a searching engine on digital documents using MR. The detailed architecture is shown in Figure 1 and here we discuss individual components, their work and results. We assume that the data has already been crawled in HDFS and the data is in the heterogeneous formats. Results returned by the proposed framework will be stored back to HDFS which can then be searched intelligently and easily. Our proposed framework consists of 3 modules based on MR three phases: *map partition*, and *reduce* modules. Each module has other sub components.

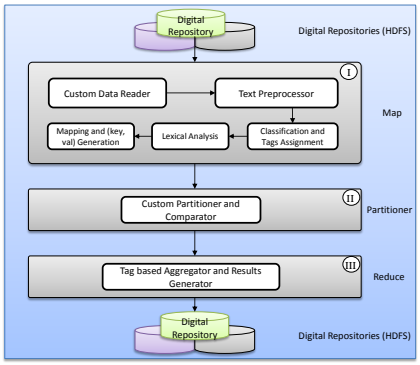

Fig. 1. Architecture of Semantic Based Search in Large Repositories

2.1 Map module

Map module shown in Figure 1 (I) shows sub-components that extract the meta data from documents semantically which is described as follows.

Custom Data Reader/Pre-processor: Since Hadoop divides input data into data chunks with default size of 64MB and data is in heterogeneous formats. Therefore, firstly we read a whole single file as input to mapper to avoid the data of a single file being distributed across multiple mappers. This helps in exact identification, indexing of the document, word, and tagging relationship. Finally, the document is parsed for slang removal. As the professional documents related to research and academia often does not contain slang, however, here

it will remove the words that do not play an important role in the context identification such as *is, am, are, they, these, those, and other many such type of words*. It will focus on the key words and vocabularies that can be used in the context identification such as sections of the document and tagging the word to the section according to their importance.

Classification and Tag Assignment/Lexical Analysis: Each digital document consists of many sections e.g. research documents contains sections: title, abstract, introduction, and conclusion. It identifies each part of document and tag the words that appear under this section. It classifies text into different section based on weight of it appearing in each section. Terms are extracted and categorized as single terms and compound terms. For compound terms, it will use the stop words mechanism while single terms are directly identified. After identification, it will perform lexical analysis using Parts Of Speech(POS) technique. POS is performed with the help of thesaurus and stemming is performed to correctly identify the concept words. A term may be occurring multiple times, therefore, we include term clustering based on the document and their frequency.

Mapping and (k,v) pair Generation: In this sub-component, results of previous are processed to *map* each word to its section, and index the section and word both to the document. Since in MR, the *map* output must be in (key, val) pairs, therefore, data is processed to generate the pairs with words as *key* and the rest of the semantic data as its *value*. (Key, val) pair generation helps in partitioner and reducer step to perform clustering and indexing of documents.

2.2 Partitioner Module

This module has the only sub-component i.e. *partitioner and comparator* as shown in Figure 1(II). The (key, val) pairs generated by the *map* module are processed by partitioner module to shuffle the data between nodes in the cluster. Shuffling is performed based on keys (concept words). Words with same semantics are forwarded to a single reducer for further aggregation and final output generation. The partitioner component is customized to avoid data skew, in case, if skew occurs, it distributes the data to other available reducers.

2.3 Aggregator and Results Generator

Figure 1(III) shows the final component in the architecture and the data forwarded by the previous modules is collected and aggregated by *reduce* module based on their keys. As their keys represent the concept, therefore, a concept occurring in each document is aggregated with all its semantics and ranked based on its frequency and importance of the section of document. The ranking helps in returning results to a query easily. each concept is also indexed to a document it belongs to. The results of reducer are in (key, val) format and stored directly into the Hadoop repository. Once all steps are performed, data in HDFS will be

ready for querying and searching. The searching of results are of two kinds: a) online search, b) offline search. In online, the query will be directly processed by this framework and the results will be returned to the user. In offline, the data is preprocessed and the query result is directly returned from the data in HDFS.

3 Discussion and Conclusion

In this paper, we presented an architecture for semantics searching over large repositories using Hadoop-MR. MR programming model is used to perform computation and process the data in parallel over cluster of commodity hardware. Semantic techniques are used to attain focus on the correct terms identification. The proposed architecture combines distributed processing with semantics based processing to provide an in-depth search with high performance over large datasets. Currently, our focus is to develop semantics based intelligent search engine using Hadoop-MR to achieve high performance and recall. In future, we intend to extend our architecture to other data formats to extend its applications.

Acknowledgments

This work was supported by the Industrial Strategic Technology Development Program (10035348, Development of a Cognitive Planning and Learning Model for Mobile Platforms) funded by the Ministry of Knowledge Economy, Korea.

References

1. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: (Big data:the next frontier for innovation, competition and productivity)
2. Pennebaker, J.W., Francis, M.E., Booth, R.J.: Linguistic inquiry and word count: Liwc 2001. Mahway: Lawrence Erlbaum Associates **71** (2001) 2001
3. Moffat, A., Zobel, J.: Self-indexing inverted files for fast text retrieval. ACM Transactions on Information Systems (TOIS) **14** (1996) 349–379
4. IEEE-org: (Ieee digital library: http://ieeexplore.ieee.org/xplore/home.jsp)
5. ACM-Org: (Acm digital library: http://dl.acm.org/)
6. of Medicine, N.L.: (Medline: http://www.nlm.nih.gov/bsd/pmresources.html)
7. Khattaka, a.: (Context-aware search in dynamic repositories of digital documents)
8. Bonino, D., Corno, F., Farinetti, L., Bosca, A.: Ontology driven semantic search. WSEAS Transaction on Information Science and Application **1** (2004) 1597–1605
9. Rodríguez, e.a.: Determining semantic similarity among entity classes from different ontologies. (IEEE Transactions on Knowledge and Data Engineering)
10. Laclavík, M., Šeleng, M., Hluchỳ, L.: Towards large scale semantic annotation built on mapreduce architecture. (In: Computational Science–ICCS 2008) 331–338
11. Ghemawat, S., Gobioff, H., Leung, S.T.: The google file system. In: ACM SIGOPS Operating Systems Review. Volume 37., ACM (2003) 29–43
12. Borthakur, D.: (Facebook has the worlds largest hadoop cluster)
13. Yuan, P., Sha, C., Wang, X., Yang, B., Zhou, A., Yang, S.: Xml structural similarity search using mapreduce. In: Web-Age Information Management. Springer (2010)
14. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Communications of the ACM **51** (2008) 107–113